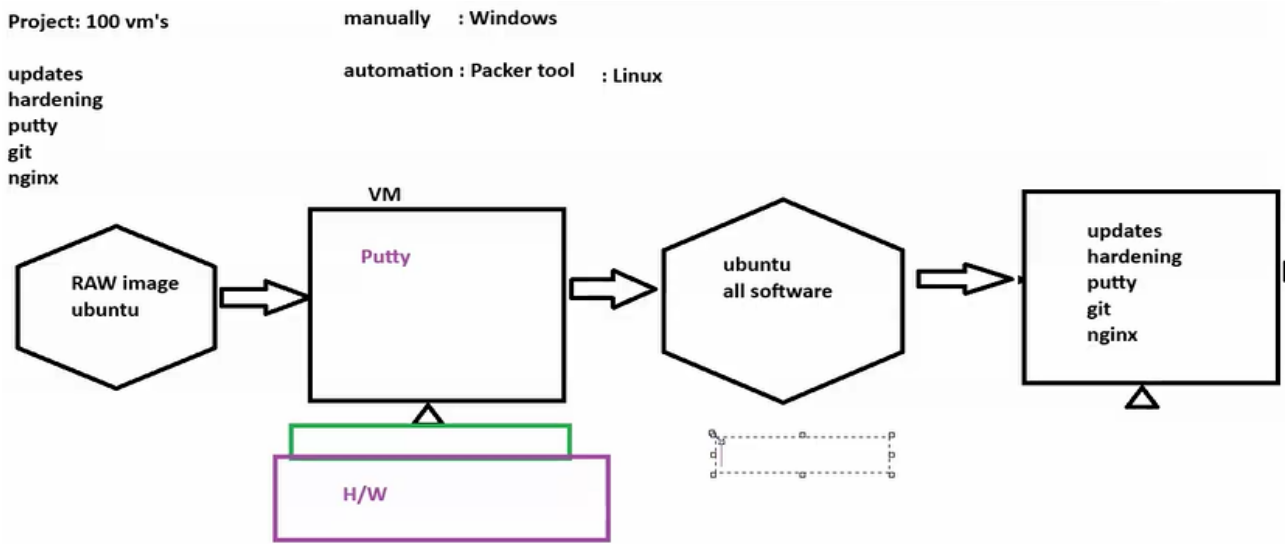
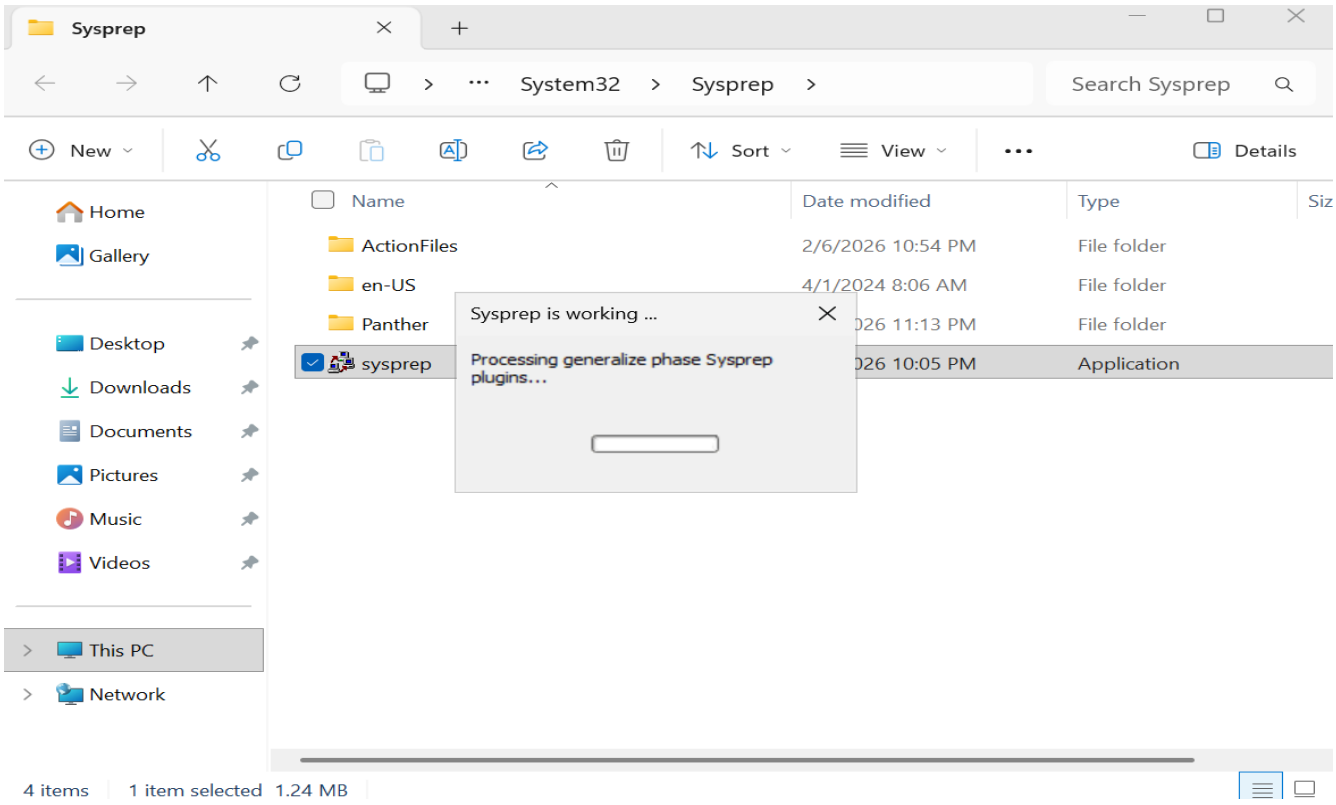
# Image Creation




- ✓ **Create Image Manually using windows**
- ✓ **Next login it and download any software (Putty) and install it.**
- ✓ **Next search for Run to shut down hardware components for custom image.**
- ✓ **Run→sysprep→select sysprep (system preparation tool)→shutdown**




- ✓ **Next go to Vm on portal we can see Vm from Running to stopped.**
- ✓ **Next click on Capture on same Vm top middle→select Create image**

# Create an image

Subscription: Azure subscription 1

Resource group *: Urgo1

**Instance details**

Region: (US) Central US

Share image to Azure compute gallery: (•) Yes, share it to a gallery as a VM image version. ( ) No, capture only a managed image.

Automatically delete this virtual machine after creating the image: [✓]

**Gallery details**

Target Azure compute gallery *: (new) Imagegallery

Create new

Operating system state: (•) Generalized: VMs created from this image require hostname, admin user, and other VM related setup to be completed on first boot ( ) Specialized: VMs created from this image are completely configured and do not require parameters such as hostname and admin user/password

⚠ Capturing a virtual machine image will make the virtual machine unusable. This action cannot be undone.

Target VM image definition *: (new) Vmdef

Create new

**Version details**

Version number *: 01.22.33

Review + create | < Previous | Next : Tags >

✓ **Custom image is created Manually by using Windows below.**

Home › Microsoft.Compute-CaptureVM-20260211161502 | Overview

## 01.22.33 (Imagegallery/Vmdef/01.22.33)

VM image version

How do I troubleshoot issues with this VM image version? | What are the metrics for this VM image version? | +1

Search

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Automation
Help

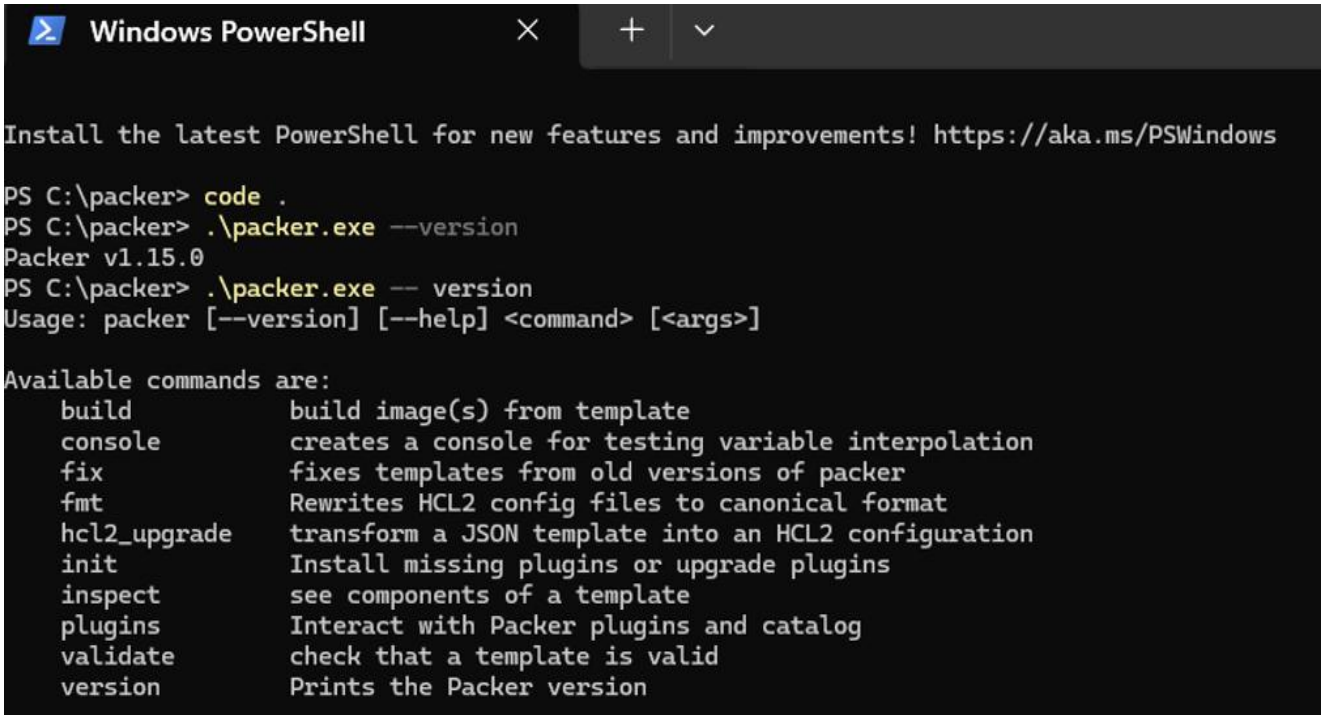
Create VM | Create VMSS | Delete | Refresh | Give feedback

**Essentials** JSON View

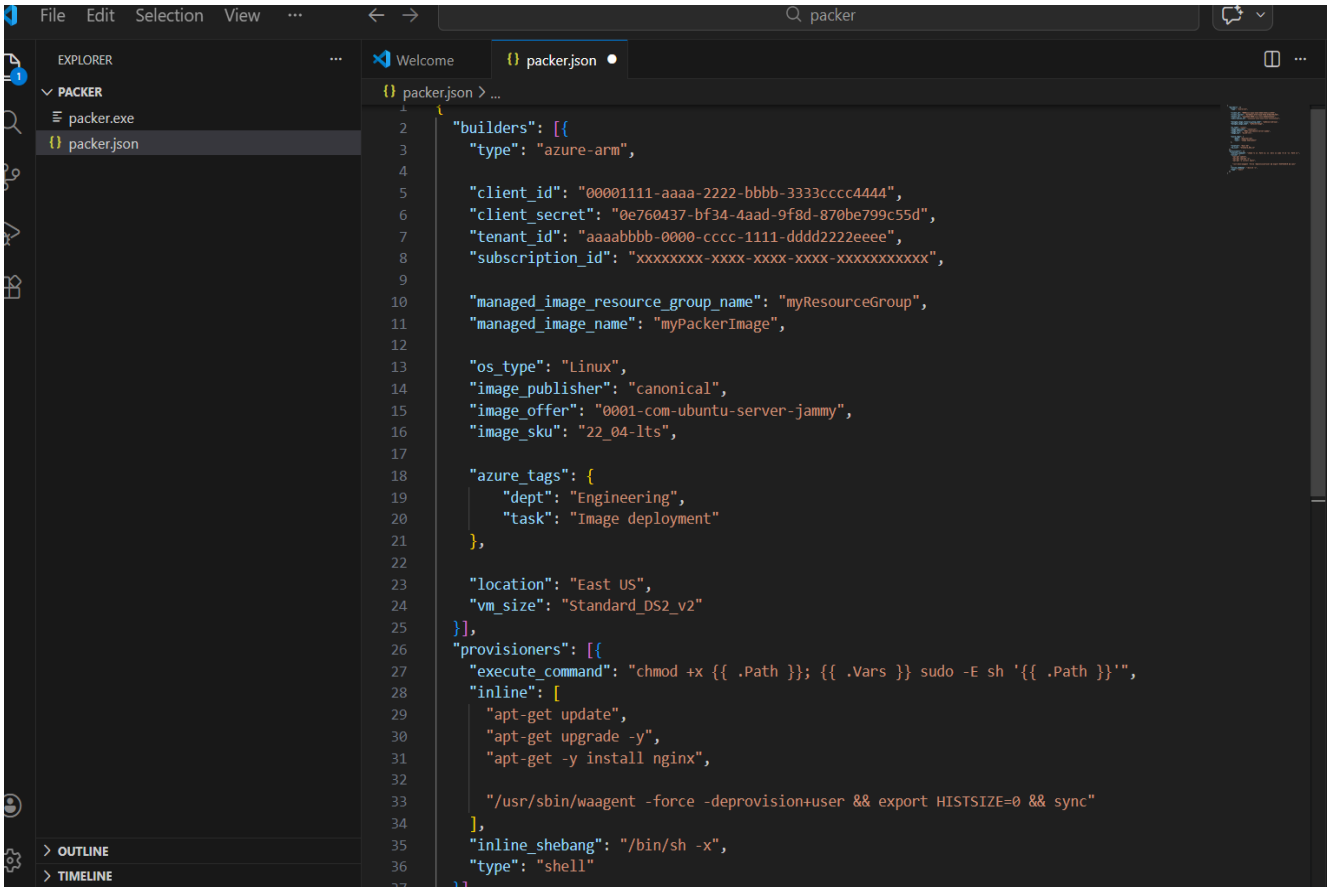
| | | | |
|---|---|---|---|
| Resource group (move) | : Urgo1 | Azure compute gallery | : Imagegallery |
| Status | : Succeeded | VM image definition | : Vmdef |
| Location | : Central US | Replication status | : Completed |
| Subscription (move) | : Azure subscription 1 | Replication mode | : Full |
| Subscription ID | : 5d75b66d-66bf-44e0-8d7e-7e61e4b043d7 | Confidential OS disk encr... | : - |
| | | Encryption type | : Platform-managed key |
| | | End of life date | : - |
| | | Exclude from latest | : No |
| | | Lock deleting Replicated ... | : Yes |
| | | Storage account type | : Standard HDD LRS |

Tags (edit) : Add tags

➢ **Let's automate same process by using Packer tool and create Custom image of Linux machine**




✓ **Code . it will navigate to VS Code**



- ✓ In Vs code there is folder packer.exe
- ✓ Click on it and choose new file and give name packer.json
- ✓ File should be in Json format only.
- ✓ Next for image creation code go to below and choose Linux format

Microsoft Learn
https://learn.microsoft.com › azure › linux › imaging

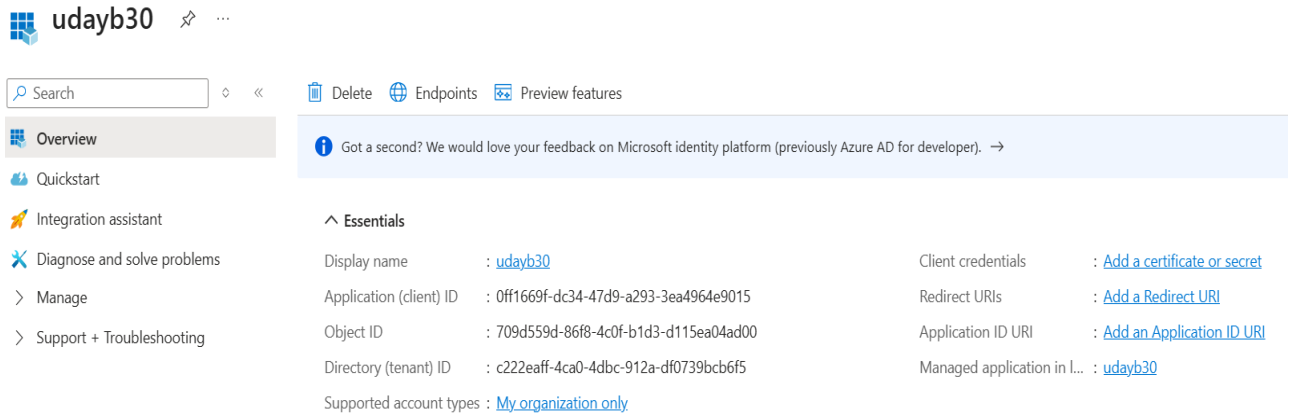
## Overview of creating Linux images for Azure

**Create an image that can be configured for use by multiple VMs**. You can set the hostname, add an admin user, and perform other tasks during first boot. Read more

- ✓ Copy Linux code and paste on VS code

```
packer => image creation

to create service account

navigate to Azure cloud => search for Microsoft entra id => manage => app registration => New registration
```

- ✓ Next Create service account for that follow steps above
- ✓ After creating account, we will get below details
- ✓ Choose client_id, client_secret, tenant_id, subscription_id copy these from below portal and paste in VS code.

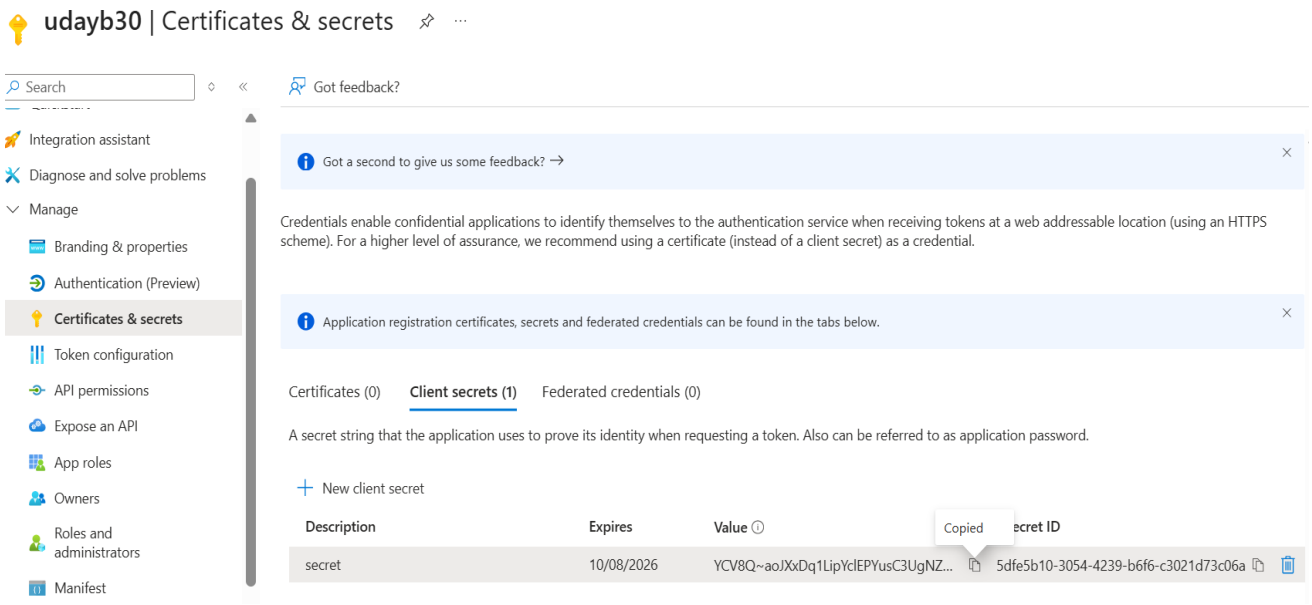



- ✓ Shown below

```
"client_id": "0ff1669f-dc34-47d9-a293-3ea4964e9015",
"client_secret": "YCV8Q~aoJXxDq1LipYclEPYusC3UgNZM-PkMjbFa",
"tenant_id": "c222eaff-4ca0-4dbc-912a-df0739bcb6f5",
"subscription_id": "5d75b66d-66bf-44e0-8d7e-7e61e4b043d7",
```

- ✓ For client_secret go to manage→certificates & secrets follow below.

✓ **Client secret id below (Copy Value)**

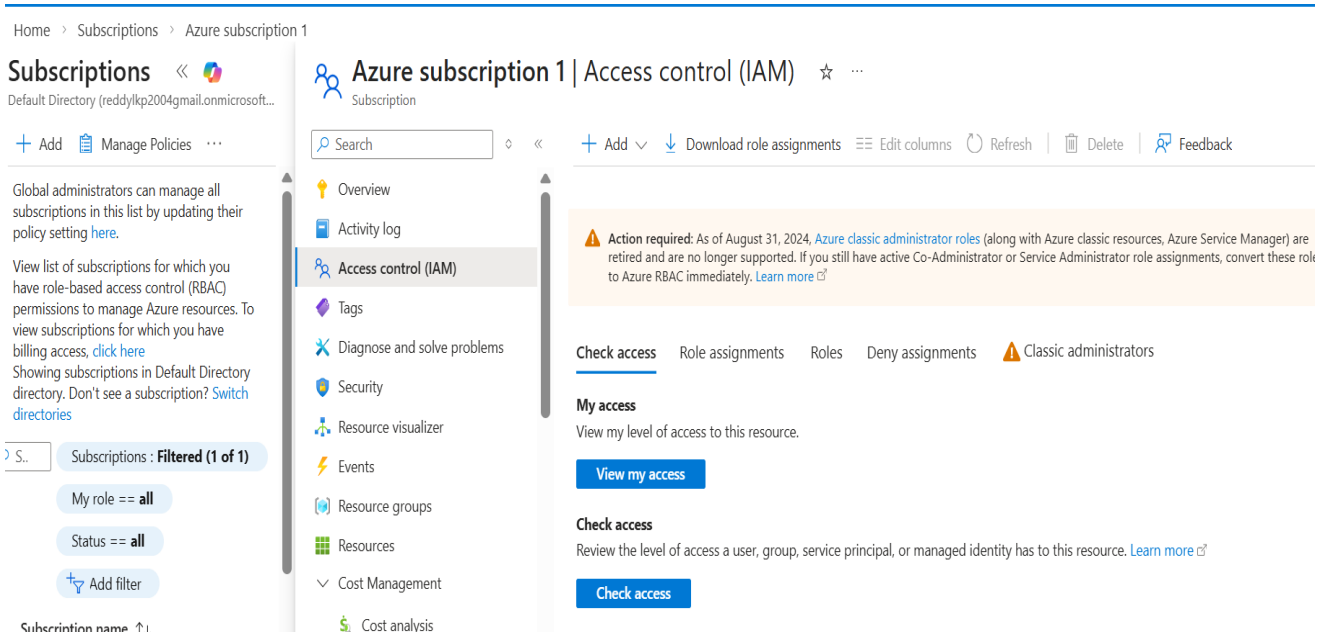


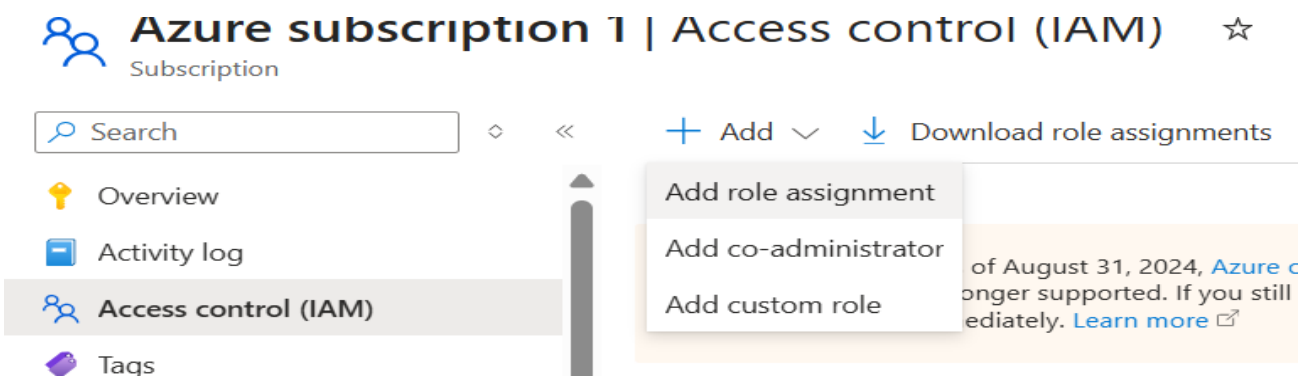

✓ **Next our free subscription ID copy and paste.**

➢ **Right now, Authentication is completed**

- **But Udayb30 user need Authorisation on subscription, for that we should give some role like Owner or etc...**

✓ **Go to IAM (Identity Access Management) below**






## Add role assignment

Role | Members | Conditions | Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Le

Copilot can help pick a role

Job function roles | **Privileged administrator roles**

Grant privileged administrator access, such as the ability to assign roles to other users.

Can a job function role with less access be used instead?

Search by role name, description, permission, or ID | Type : **All** | Category : **All**

| Name | Description |
|---|---|
| Owner | Grants full access to manage all resources, including the ability to assign r |
| Contributor | Grants full access to manage all resources, but does not allow you to assig |
| Access Review Operator Service Role | Lets you grant Access Review System app permissions to discover and rev |
| Azure IoT Operations Onboarding | User can Azure arc connect and deploy Azure IoT Operations securely. |
| Azure Resilience Management Drills Admi... | Administrator Role of Azure Resilience Management Drills Service |

Review + assign | Previous | Next

✓ **Below added user Udayb30 and I want to give owner role**

Role | **Members** | Conditions | Review + assign

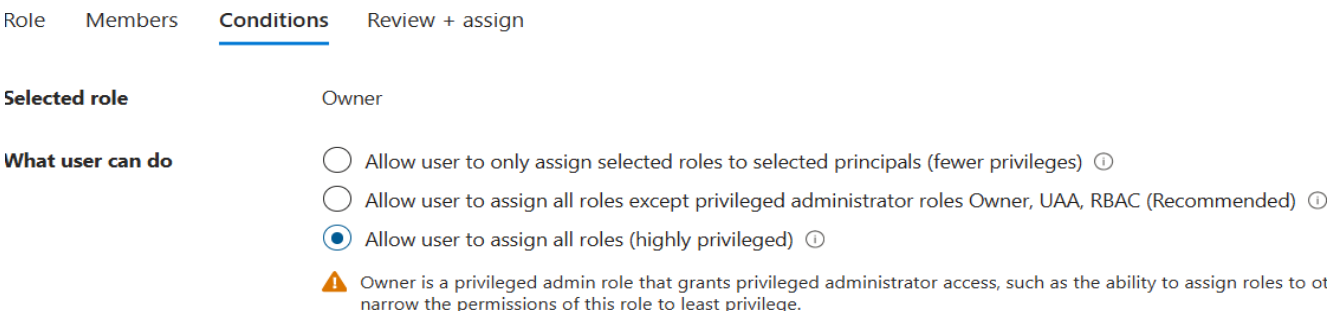
**Selected role** Owner

**Assign access to**
- (●) User, group, or service principal
- ( ) Managed identity

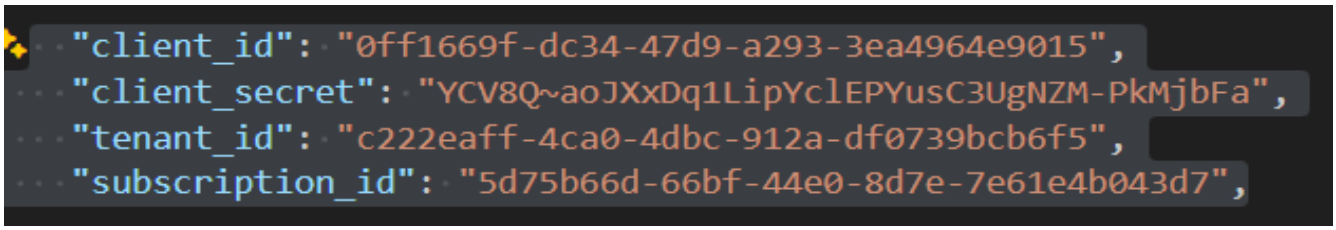
**Members** + Select members

| Name | Object ID | Type |
|---|---|---|
| udayb30 | f28ed8b3-ef51-414b-bb06-69eff71a4b96 | App |

**Description** Optional




✓ **To run packer, we need Authentication with authorisation then only we can execute, we have done all below.**




✓ **Below give**

- **Resource group name**
- **Image name**
- **OS type**
- **Location**
- **VM size**

Welcome | packer.json

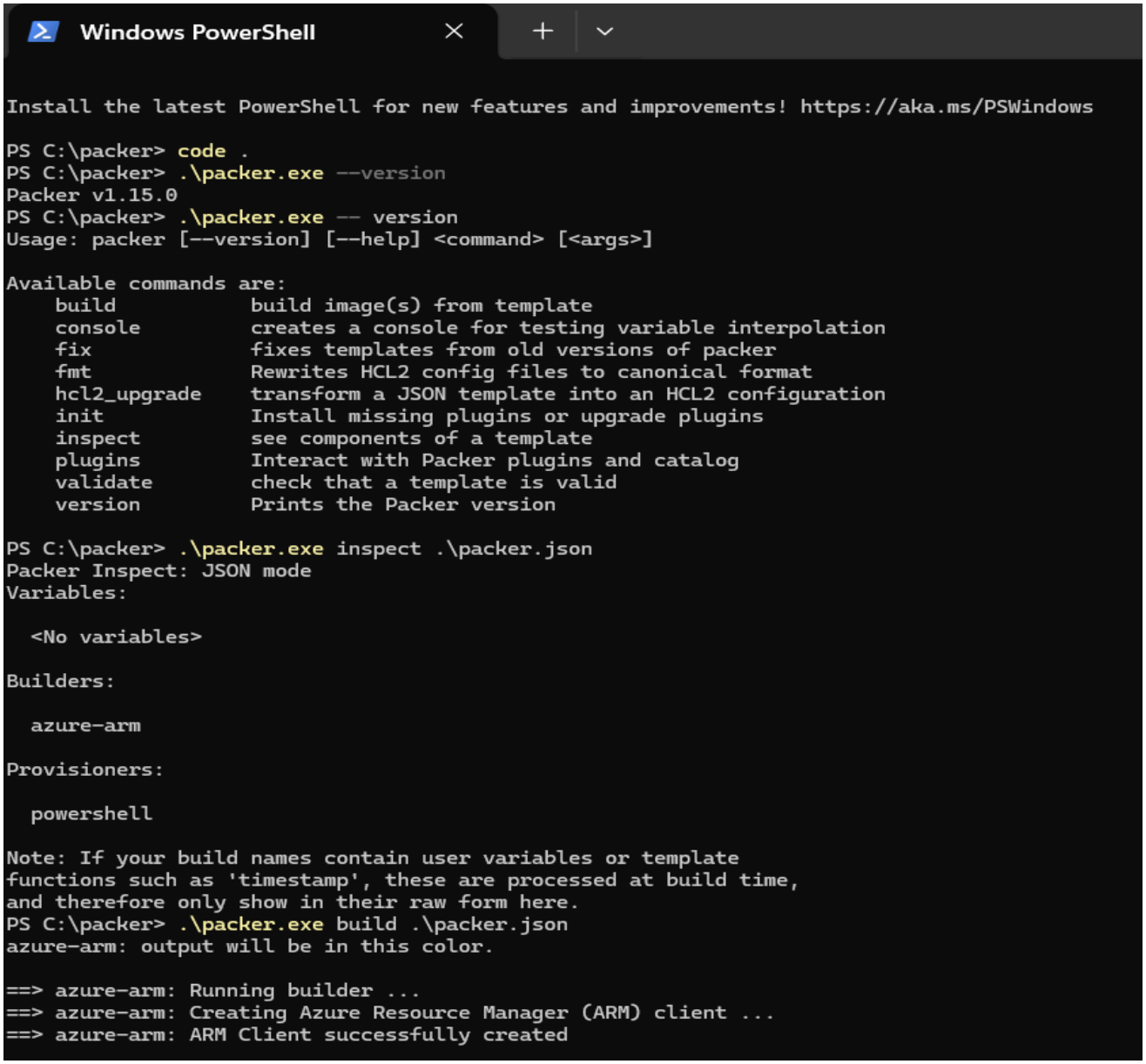
packer.json > [ ] builders > { } 0 > abc vm_size

```
  "builders": [{
    "type": "azure-arm",

    "client_id": "0ff1669f-dc34-47d9-a293-3ea4964e9015",
    "client_secret": "YCV8Q~aoJXxDq1LipYclEPYusC3UgNZM-PkMjbFa",
    "tenant_id": "c222eaff-4ca0-4dbc-912a-df0739bcb6f5",
    "subscription_id": "5d75b66d-66bf-44e0-8d7e-7e61e4b043d7",

    "managed_image_resource_group_name": "Uday-Rg",
    "managed_image_name": "myudaylinux-Image",

    "os_type": "Linux",
    "image_publisher": "canonical",
    "image_offer": "0001-com-ubuntu-server-jammy",
    "image_sku": "22_04-lts",

    "azure_tags": {
        "dept": "Engineering",
        "task": "Image deployment"
    },

    "location": "East US",
    "vm_size": "Standard_B1s"
  }],
  "provisioners": [{
    "execute_command": "chmod +x {{ .Path }}; {{ .Vars }} sudo -E sh '{{ .Path }}'",
    "inline": [
      "apt-get update",
      "apt-get upgrade -y",
      "apt-get -y install nginx",

      "/usr/sbin/waagent -force -deprovision+user && export HISTSIZE=0 && sync"
    ],
    "inline_shebang": "/bin/sh -x",
    "type": "shell"
  }]
```

- ✓ **Make sure everything is done save first Ctrl+s**
- ✓ **Then open command prompt and execute below**

```
PS C:\packer> .\packer.exe inspect .\packer.json
Packer Inspect: JSON mode
Variables:

  <No variables>

Builders:

  azure-arm

Provisioners:

  shell

Note: If your build names contain user variables or template
functions such as 'timestamp', these are processed at build time,
and therefore only show in their raw form here.
PS C:\packer>
```

✓ **Below is complete code from start to build**




✓ **For the first time before packer build, we should install Plugin.**

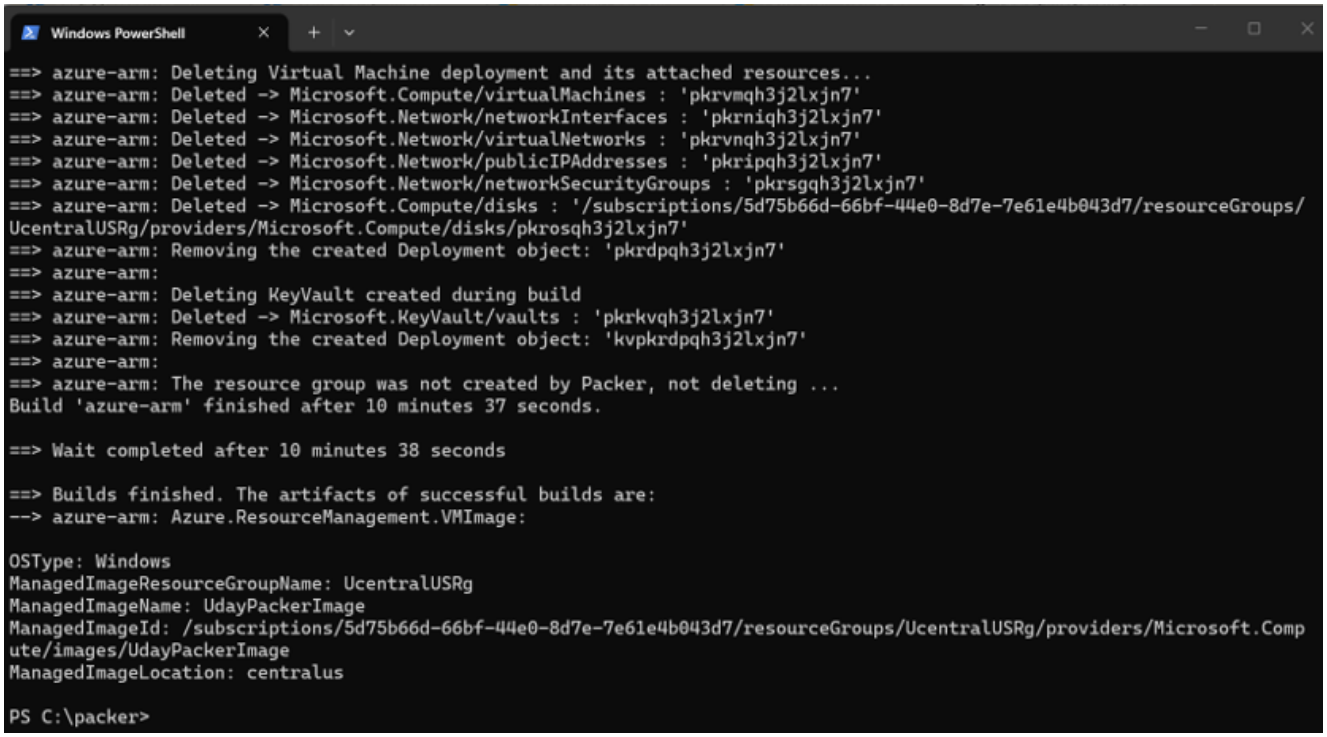
✓ **If we don't it asks to install plugin,then enter below**

✓ **./packer plugins install github.com/hashicorp/azure**

```
$ packer plugins install github.com/hashicorp/azure
```

✓ **Next run build command, packages will build automatically and custom image we see in resource group (Uday-Rg).**

✓ **Image is Created.**



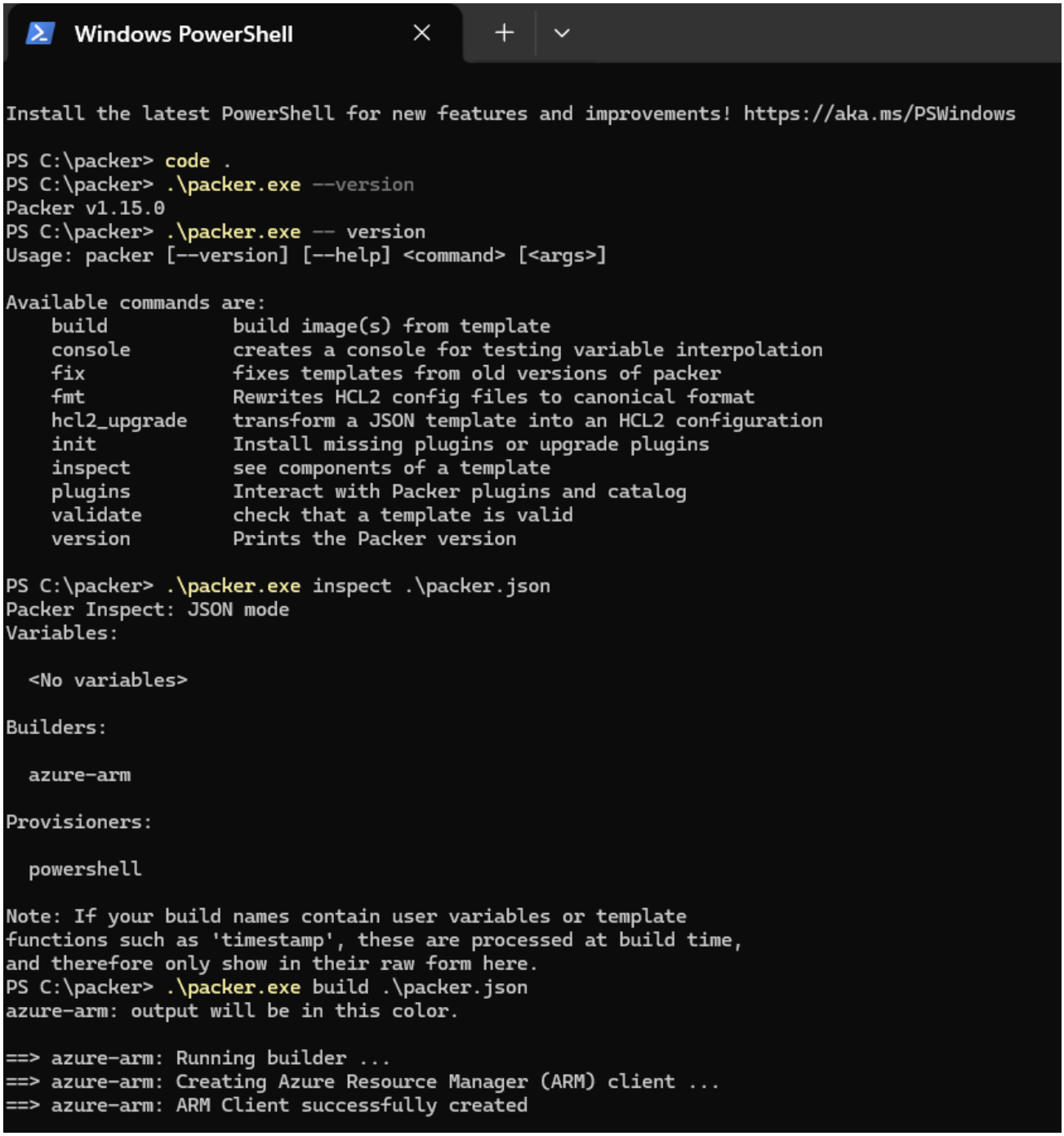

➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢
➢

## ➢ Automate Windows (Image Creation)

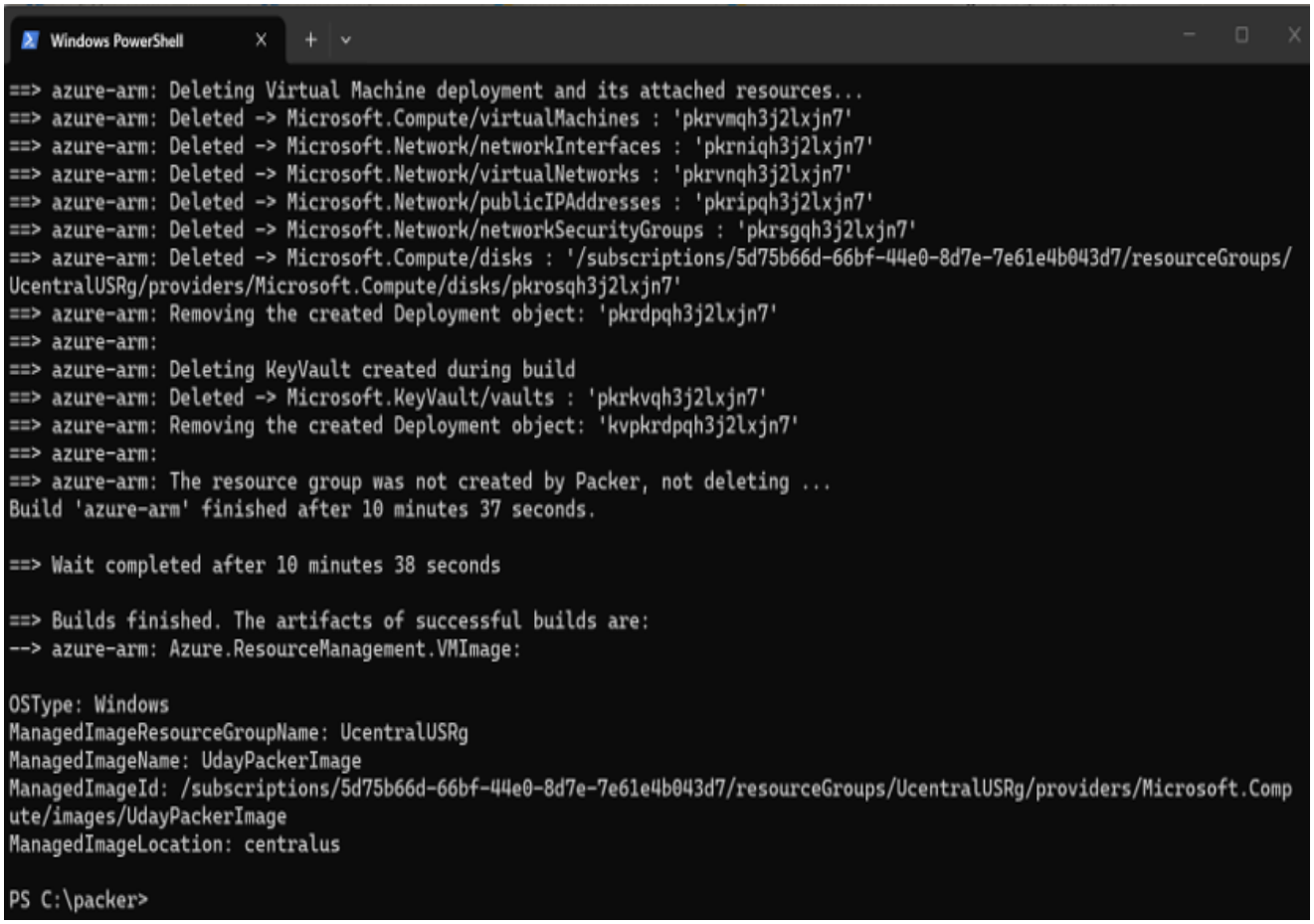
packer.exe  {} packer.json

{} packer.json > [ ] builders > { } 0

```
{
  "builders": [{
    "type": "azure-arm",

    "client_id": "bbd8d2fa-5fc7-40f5-97a4-ea93ed316618",
    "client_secret": "IxS8Q~hzGAvegg_BkxlU02nCXAg09MJge4mapbZP",
    "tenant_id": "c222eaff-4ca0-4dbc-912a-df0739bcb6f5",
    "subscription_id": "5d75b66d-66bf-44e0-8d7e-7e61e4b043d7",

    "managed_image_resource_group_name": "UcentralUSRg",
    "managed_image_name": "UdayPackerImage",

    "os_type": "Windows",
    "image_publisher": "MicrosoftWindowsServer",
    "image_offer": "WindowsServer",
    "image_sku": "2016-Datacenter",

    "communicator": "winrm",
    "winrm_use_ssl": true,
    "winrm_insecure": true,
    "winrm_timeout": "5m",
    "winrm_username": "packer",

    "azure_tags": {
        "dept": "Engineering",
        "task": "Image deployment"
    },

    "build_resource_group_name": "UcentralUSRg",
    "vm_size": "Standard_D2s_v3"
  }],
  "provisioners": [{
    "type": "powershell",
    "inline": [
      "Add-WindowsFeature Web-Server",
      "while ((Get-Service RdAgent).Status -ne 'Running') { Start-Sleep -s 5 }",
      "while ((Get-Service WindowsAzureGuestAgent).Status -ne 'Running') { Start-Sleep -s 5 }",
      "& $env:SystemRoot\\System32\\Sysprep\\Sysprep.exe /oobe /generalize /quiet /quit",
      "while($true) { $imageState = Get-ItemProperty HKLM:\\SOFTWARE\\Microsoft\\Windows\\Current
    ]
  }]
}
```




- ✓ **Now we no need to install plugin just run build package will be created.**


Windows PowerShell
==> azure-arm: Deleting Virtual Machine deployment and its attached resources...
==> azure-arm: Deleted -> Microsoft.Compute/virtualMachines : 'pkrvmqh3j2lxjn7'
==> azure-arm: Deleted -> Microsoft.Network/networkInterfaces : 'pkrniqh3j2lxjn7'
==> azure-arm: Deleted -> Microsoft.Network/virtualNetworks : 'pkrvnqh3j2lxjn7'
==> azure-arm: Deleted -> Microsoft.Network/publicIPAddresses : 'pkripqh3j2lxjn7'
==> azure-arm: Deleted -> Microsoft.Network/networkSecurityGroups : 'pkrsgqh3j2lxjn7'
==> azure-arm: Deleted -> Microsoft.Compute/disks : '/subscriptions/5d75b66d-66bf-44e0-8d7e-7e61e4b043d7/resourceGroups/
UcentralUSRg/providers/Microsoft.Compute/disks/pkrosqh3j2lxjn7'
==> azure-arm: Removing the created Deployment object: 'pkrdpqh3j2lxjn7'
==> azure-arm:
==> azure-arm: Deleting KeyVault created during build
==> azure-arm: Deleted -> Microsoft.KeyVault/vaults : 'pkrkvqh3j2lxjn7'
==> azure-arm: Removing the created Deployment object: 'kvpkrdpqh3j2lxjn7'
==> azure-arm:
==> azure-arm: The resource group was not created by Packer, not deleting ...
Build 'azure-arm' finished after 10 minutes 37 seconds.
==> Wait completed after 10 minutes 38 seconds
==> Builds finished. The artifacts of successful builds are:
--> azure-arm: Azure.ResourceManagement.VMImage:
OSType: Windows
ManagedImageResourceGroupName: UcentralUSRg
ManagedImageName: UdayPackerImage
ManagedImageId: /subscriptions/5d75b66d-66bf-44e0-8d7e-7e61e4b043d7/resourceGroups/UcentralUSRg/providers/Microsoft.Comp
ute/images/UdayPackerImage
ManagedImageLocation: centralus
PS C:\packer>